УДК 681.3.068

ПОДСИСТЕМА ВИЗУАЛИЗАЦИИ НАУЧНЫХ ДАННЫХ ДЛЯ СИСТЕМНОЙ ОБОЛОЧКИ BRAINSTORM

В.П. Зимин, Е.А. Хомяков

Томский политехнический университет
E-mail: zimin@ido.tpu.edu.ru

Обсуждается проблема визуализации научных данных при проведении вычислительных экспериментов с помощью пакетов прикладных программ, поддерживаемых системной оболочкой BrainStorm, разработанной в ТПУ. Предлагается ее решение, состоящее в разработке специализированных модулей визуализации, позволяющих отображать и анализировать в графическом виде данные как во время проведения численных экспериментов, так и после их завершения.

Визуализация научных данных является важным и необходимым этапом, а так же актуальной задачей процесса интерпретации результатов как физического, так и вычислительного экспериментов. Актуальность данной задачи значительно возрастает при необходимости интеграции отдельных программ в пакеты прикладных программ (ППП). Эта задача может быть решена, по крайней мере, двумя способами. Первый – это использование подсистемы визуализации каких-либо пакетов программ, например, Mathcad или Statistica. Второй способ – это программирование отрисовки всех деталей графиков зависимостей, возможно, с использованием имеющихся графических библиотек.

Имеющиеся подсистемы визуализации математических пакетов и графические библиотеки визуализации позволяют эффективно решать задачу графического представления разнообразных зависимостей. Но при интеграции этих подсистем визуализации с системным и функциональным обеспечением вновь создаваемых ППП возникают определенные трудности. Их анализ показывает, что большая часть вычислительных возможностей имеющихся подсистем визуализации (или их компонент) оказывается избыточной, а связанная с этим внутренняя логика и способы визуализации и обмена данными представляют собой препятствие для интеграции программ, иногда – практически непреодолимое. В результате уникальные подсистемы визуализации универсальных пакетов становятся недоступными разработчику ППП. Например, чтобы получить доступ к графическим возможностям специализированного пакета, требуется создавать файл в специальном формате, который этот пакет сможет прочитать. Затем команда «Импортировать», в том или ином виде присутствующая во многих пакетах, позволяет пакету прочитать этот файл и добавить в собственный документ данные,

графики и т. д. В некоторых случаях разработчики специально ориентируют свои программы на совместную работу с каким-либо широко распространенным пакетом, т.е. фактически расширяют его функциональность. Но в большинстве случаев один и тот же набор проблем практически (в разных комбинациях) заставляет отказаться от взаимодействия со специализированным пакетом. Перечислим некоторые из таких проблем:

- Сложные форматы импорта. Все интерпретируемые пакетом форматы настолько сложны, что цена добавления в свои модули возможности писать в этих форматах может в иных случаях превысить затраты на разработку собственного простого модуля визуализации.
- Необходимость дистрибуции специализированного пакета. Не всегда модули, разрабатываемые исследователем, используются только им самим. Очень часто они в том или ином виде передаются другим людям – ученым или конечным пользователям. Если в случае простой передаче дружественной группе исследователей это не составит проблемы, то в случае коммерческого распространения почти во всех случаях просто невозможно распространять специализированный пакет, например, из-за лицензионных ограничений или его цены.
- Ограниченность специализированного пакета. Некоторые, замечательные с точки зрения пользователя, пакеты не имеют какого-либо даже самого примитивного интерфейса программного управления, или даже не позволяют открыть/импортировать файл, передавая его имя через командную строку.

В свою очередь, графические библиотеки визуализации данных для разных языков программирования, помимо очевидного недостатка – фактической доступности только из одного, «родного» языка – имеют еще один, неочевидный недостаток: необходимость выбора между простотой использования и его функциональностью. В подавляющем большинстве случаев выбирается последнее. Тогда как наиболее часто, особенно на этапе первичной интерпретации результатов вычислительного (физического) эксперимента, требуется максимально простым и быстрым способом изобразить набор графиков на плоскости или гистограмм для массивов чисел. В то же время на этапе исследования и оформления результатов эксперимента должна оставаться возможность модифицировать (редактировать) это множество графиков или гистограмм.

Опыт создания системной оболочки BrainStorm v. 1.0 под операционной системой (ОС) MS Windows для поддержки унаследованных ППП [1] и опыт модификации с ее помощью пакета моделирования процессов в термоэмиссионных системах [2] показал, что необходимо существенно обновить функциональные модули визуализации. Поэтому подсистема визуализации вновь создаваемых пакетов состоит из двух уровней. На первом уровне модули визуализации пакета формируют файлы визуализации, которые имеют XML-структуру. Файлы визуализации содержат входные и выходные данные функциональных модулей, которые необходимо визуализировать, а так же структуру их представления и компоновку в виде графиков. На втором уровне находится программа визуализации BsVis, которая использует файл визуализации как параметр командной строки. Организованные таким образом модули визуализации пакета могут выполняться как обычные функциональные (вычислительные) модули пакета, поддерживаемого BrainStorm, в двух режимах: в интерактивном и пакетном [1]. В первом случае, после формирования файла визуализации он сразу интерпретируется с помощью запуска программы BsVis непосредственно из модуля визуализации. Во втором случае создается в банке данных файл визуализации, а его интерпретация BsVis отложена во времени. Формировать файл визуализации можно различными способами: непосредственно в функциональном (вычислительном) модуле, с помощью текстового редактора или подстановкой данных в заранее подготовленный шаблон.

Основное назначение программы визуализации BsVis – статическая визуализация массивов чисел или пар чисел в виде графиков (точки соединяются отрезками прямых), гистограмм, кластеров, векторов и т.п. на плоскости. Пользователю наиболее удобно проводить анализ количественных соотношений между данными на плоскости, тогда как в пространстве наиболее успешна качественная интерпретация данных. При написании данной программы была использована графическая компонента TeeChart 7.7 для системы визуального программирования Delphi [3].

Таким образом, программа BsVis решает следующие проблемы графических библиотек и модулей визуализации специализированных пакетов:

- Доступна практически из любой среды программирования, точнее из тех, которые позволяют запустить программу, разработанную под ОС MS Windows. Имя файла данных принимается из командной строки.
- Использует простой, стандартизированный международным консорциумом W3C формат XML. Подмножество XML, использующееся для представления файла визуализации, следует принципу – чем проще действие, которое требуется совершить, тем меньшее количество инструкций необходимо указать.
- Накладывая ограничения на функциональность, она исключает необходимость выполнять любые неочевидные действия при формировании файла визуализации. Например, чтобы расставить точки в узлах графической зависимости, исследователь просто дает команду отобразить точки, вместо того, чтобы создавать объект-расширение системы рисования «`Расстановщик точек в узлах`» и подключать

его к системе визуализации. Это то, что делают многие специализированные пакеты – предоставляют ограниченный нерасширяемый набор настроек графика, который на самом деле почти всегда достаточен. Причем настройки по умолчанию подобраны так, чтобы их как можно реже требовалось изменять – т. е. наиболее часто используемые настройки приняты в качестве стандартных.

Программа BsVis имеет два режима визуализации: простой (ординарный) и расширенный. В ординарном режиме работе программа интерпретирует содержимое файла визуализации. Расширенный режим визуализации доступен с помощью контекстного меню рисунка и позволяет: видоизменять представление рисунка и отдельных графиков на нем; сохранять данные изменения в виде нового XML-файла; организовывать исследование полученных графических данных; копировать рисунок в буфер и т. д.

Основой ординарного режима является файл визуализации. Как и любые XML-документы, файлы, полученные с помощью модуля визуализации, характеризуются содержанием, структурой (разметкой) и внешним представлением. Содержание файлов визуализации, отображающее частично или полностью входные и выходные данные функциональных (вычислительных) модулей, окончательно интерпретируется пользователем, проводящим вычислительный (физический) эксперимент. Эта интерпретация основывается на представлении XML-файла программой BsVis. В свою очередь разметка (структура) файла правильно определяет составляющие его части и взаимоотношение между ними.

В общем случае файл визуализации имеет шесть иерархически вложенных элементов, причем пять из них характеризуются несколькими атрибутами. Корневым является элемент `<vis>`. Он атрибутов не имеет и содержит элемент `<page>` (2 атрибута). Данный элемент представляет описание одной страницы визуализации и включает элемент `<chart>` (3 атрибута), описывает один рисунок. Для каждого рисунка определяется несколько элементов. В первую очередь два элемента `<xaxis>` и `<yaxis>` (по 6 атрибутов в каждом), определяет структуру осей рисунка. Элемент `<series>` (6 атрибутов) описывает представление одной функции (или гистограммы) на рисунке. И, наконец, шестой элемент `<pointer>` (5 атрибутов) позволяет изображать график в виде точек разных стилей. В заголовке файла определяется кодовая страница символов файла, например, для унаследованного программного обеспечения – CP866, для Windows-приложений – CP1251 и т. д.

Таким образом, можно визуализировать несколько страниц, организуемых в виде закладок; на одной странице может быть расположено до четырех рисунков, на одном рисунке – несколько графиков функции. Каждая функция описывается набором пар действительных чисел, гистограмма – массивом чисел. График представляется либо линией (ломаной кривой, состоящей из отрезков прямых), либо точками в узлах функции. Гистограмма представляется в виде столбчатой диаграммы. Для каждых графика функции и гистограммы можно задавать тип, стиль, цвет и размер.

Например, разработчику требуется визуализировать графики функций и гистограмму (одномерный массив чисел): y_i – 65,70; 103,00; 144,60; 87,90; 96,15; 132,90. Тогда вместо подключения компонентов и их программирования, достаточно для визуализации отдельно гистограммы создать файл визуализации (`gist.xml`):

```
<vis><chart><series type=«bar»>
65.7 103.0 144.6 87.9 96.15 132.9
</series></chart></vis>
```

и запустить модуль визуализации: bsvis.exe gist.xml. На рис. 1 представлено совместная визуализация нескольких графиков и гистограммы при включении некоторых атрибутов элементов, описанных выше.

Для перехода в расширенный режим визуализации вызывается контекстное меню рисунка:

```
Копировать как рисунок
Универсальный редактор графиков...
Трассировка...
Сохранить как...
```

С помощью опции `Копировать как рисунок` образ рисунка помещается в буфер ОС MS Windows и впоследствии может быть помещен, например, в документ MS Word.

С помощью меню `Универсальный редактор графиков...` (рис. 2) можно изменить вид рисунка (надписи, вид графика, легенды и т. п.). Кроме этого можно просматривать данные графика в цифровом виде, подключать дополнительные инструменты к универсальному редактору, экспортировать рисунок в разных форматах и печатать рисунок.

Например, тип линии графика меняется с помощью форматирования границ серии: `Серии → (номер кривой)→ Формат → Граница.`

Тип точек графика меняется с помощью меню:

`Серии → (номер кривой) → Точка.`

Изменение вида графика – с помощью экранной кнопки `Изменить....`

При анализе графиков удобно анализировать сочетания различных графиков на одном рисунке. Это можно достичь с помощью опций `Диаграмма`:

`Диаграмма → Легенда → (No check boxes, Кнопка выбора, Radio buttons).`

Опция `Трассировка...` позволяет провести исследование как графиков, так и рисунка в целом – получить значения координат любой его точки. С помощью установки флага `Привязать к графику трассировка` будет производится только по точкам графика. Корректный выход из режима трассировки – нажатие на кнопку `<Закрыть>`.

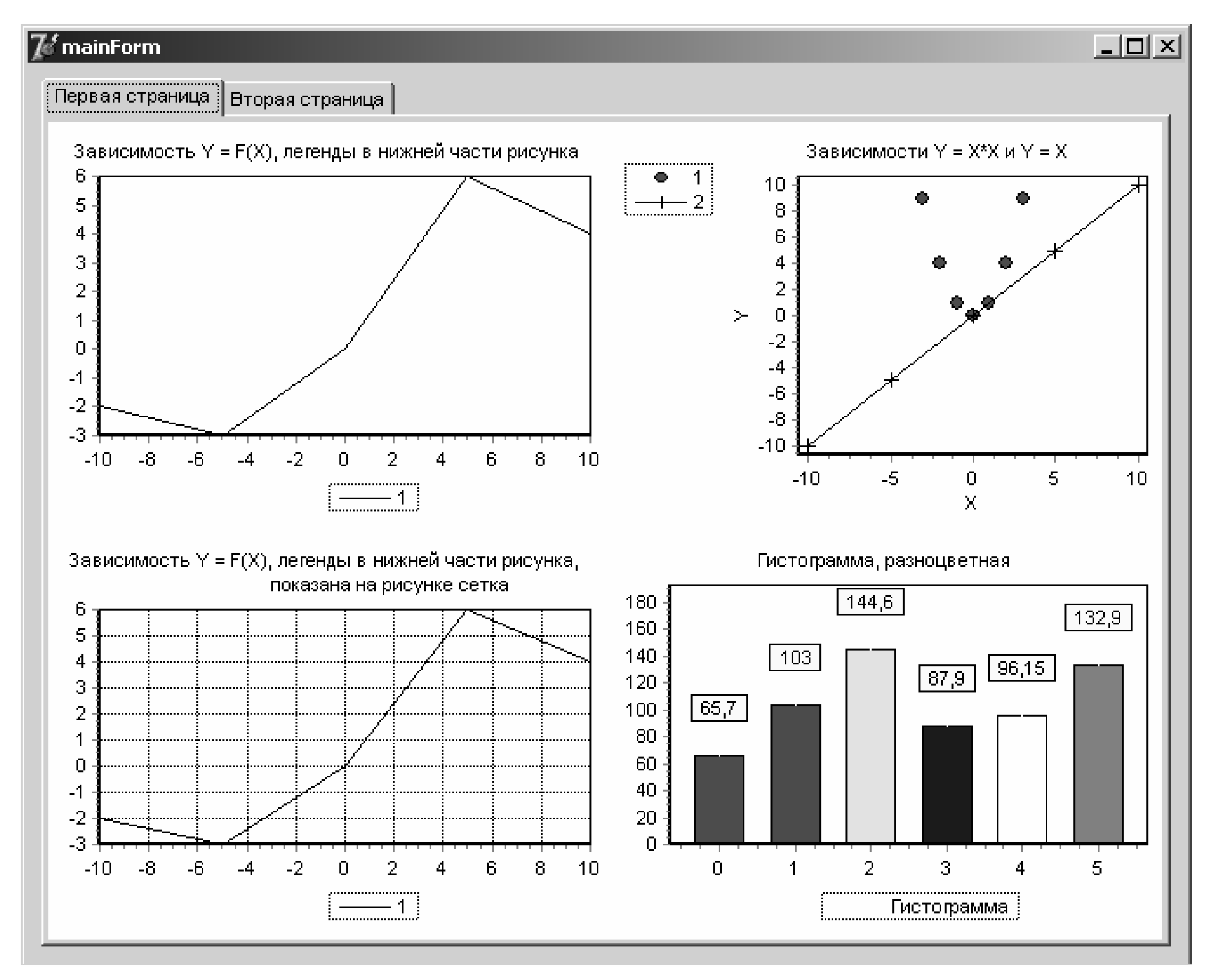


Рис. 1. *Пример визуализации графиков функции и гистограмм*

Наконец, опция `Сохранить как...` сохраняет в новом XML-файле некоторые изменения, выполненные с помощью опций меню `Универсальный редактор графиков...`.

Все это позволяет пользователю при необходимости подобрать нужный вид графика и затем перенести настройки в модуль визуализации, который формирует XML-файл, т. е. в конечном итоге – интерактивно спроектировать файл визуализации.

Кроме создания программ для подсистемы визуализации ППП, следует выделить еще одну область применения программы BsVis – в образовании, в качестве вспомогательного инструмента при выполнении студентами лабораторных, курсовых, научно-исследовательских и выпускных квалификационных работ. В большом числе случаев требуется вывод нескольких графиков зависимостей, полученных студентами в результате проведения опытов или моделирования. Обычно это решается путем использования различных графических библиотек соответствующих систем программирования. Значительное время тратится на создание множества одноразовых типовых программ, которые пишутся чтобы, например, увидеть характер процесса, и в которых выполняются однообразные действия: добавить библиотеку, создать и настроить объект-график, прочитать массив из файла и т. п. Кроме этого, требуется нескольких десятков запусков программ для подбора внешнего вида графиков. Формируя файл визуализации и используя программу BsVis, студент-исследователь сосредотачивается на творческой части процесса моделирования, уделяя минимальные время и усилия рутинным операциям.

Апробация изложенного подхода разработки подсистемы визуализации ППП использовалась при создании пакета KIAE_2004, предназначенного для моделирования процессов в термоэмиссионных системах [4]. Подсистема визуализации данного пакета состоит из программы BsVis, 15 модулей визуализации, которые формируют 23 XML-файла. Полученные графические зависимости параметров процессов в термоэмиссионных системах полностью совпали с зависимостями, ранее полученных в [2]. Необходимо отметить простоту формирования как многостраничных рисунков (компоновки рисунков на одной странице), так и простоту формирования графиков функций на одном рисунке, последующее редактирование рисунков для окончательного опубликования с помощью одного из графических редакторов.

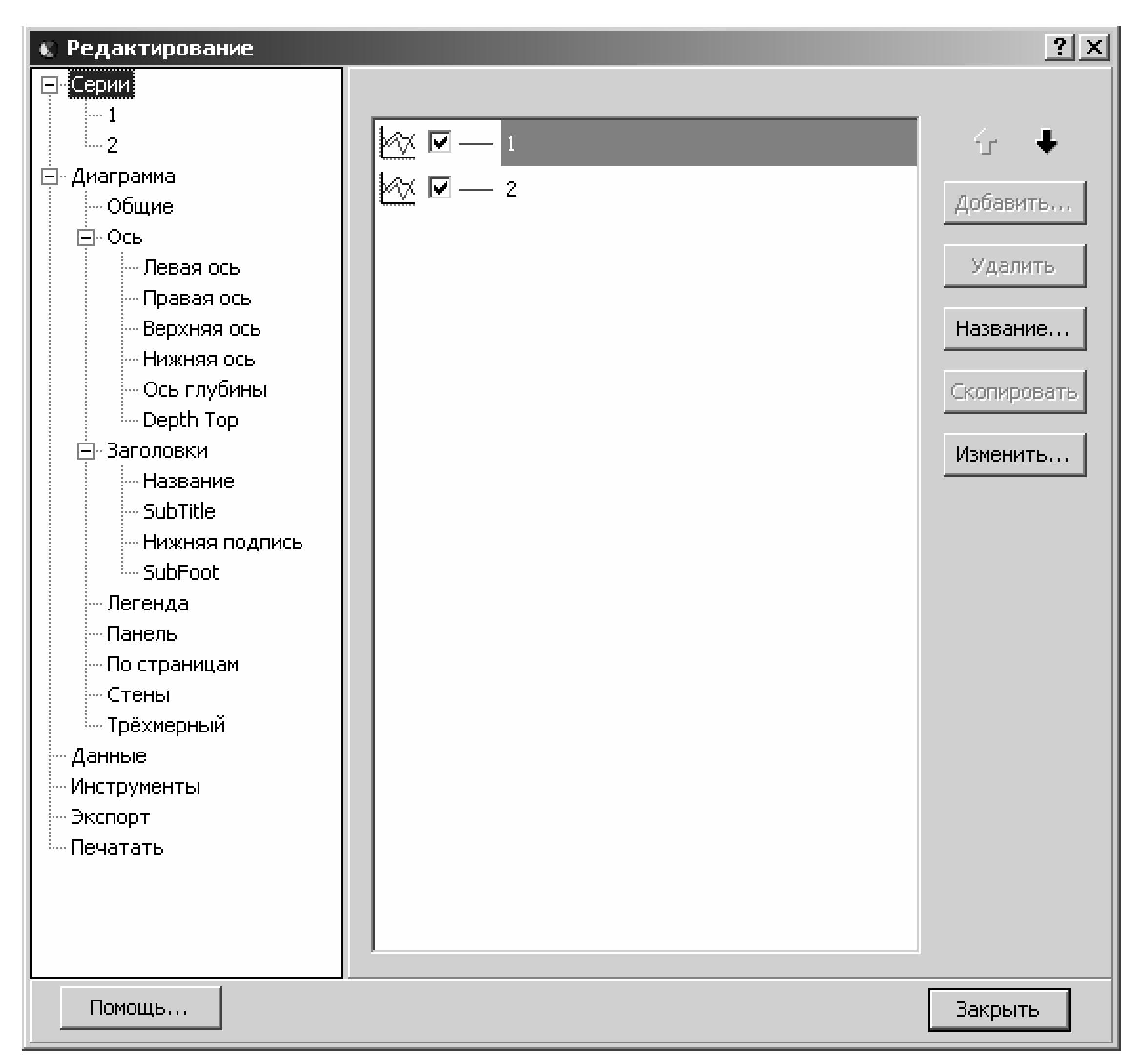


Рис. 2. *Меню универсального редактора графиков*

Таким образом, можно сделать следующие выводы:

- Реализация концепции файла визуализации, имеющего XML-структуру, и программы BsVis позволяет создавать подсистему визуализации для пакетов прикладных программ, поддерживаемых системной оболочкой BrainStorm.
- Модули визуализации, как и функциональные (вычислительные) модули вновь создаваемых пакетов, выполняются в двух режимах: интерактивном и пакетном. Это позволяет включать модули визуализации в цепочки модулей, которые исполняются в пакетном режиме.
- Разработанная программа визуализации BsVis используется для организации (интерпретации) данных как в научной работе, так и в учебном процессе.

СПИСОК ЛИТЕРАТУРЫ

1. Зимин В.П., Хомяков Е.А. Системная оболочка BrainStorm для поддержки унаследованного программного обеспечения // Известия Томского политехнического университета. – 2006. – Т. 309. – № 4. – С. 163–168.
2. Бабушкин Ю.В., Зимин В.П., Синявский В.В. Моделирующая система KOPTES для исследования тепловых и электрических процессов в термоэмиссионных системах преобразования энергии // Ракетно-космическая техника: Труды. Сер. XII. – Вып. 1–2. Расчет, проектирование, конструирование и испытания космических систем / Под ред. В.В. Синявского. – Калининград, Моск. обл.: РКК «Энергия», ОНТИ, 1998. – С. 60–78.
3. Система визуального программирования Delphi. – Режим доступа: http://www.borland.com/delphi/, вход свободный.
4. Бабушкин Ю.В., Зимин В.П., Хомяков Е.А. Программное обеспечение и результаты моделирования термоэмиссионных систем // Известия Томского политехнического университета. – 2006. – Т. 309. – № 3. – С. 53–57.